thinkdsp - CPP Companion to ThinkDSP

Srini, rs@toprllc.com

March 30, 2021

1 Introduction

This projectlet is a C++ companion to learning fundamentals of Digital Signal Processing in the tradition of ThinkDSP (`https://github.com/AllenDowney/ThinkDSP`). The goal is to be pragmatic - to achieve the objectives leveraging the numerous support libraries in almost any programming language. It is also the goal to integrate available tools judiciously. As much as possible the libraries mentioned herein will be utilized supplying a wrapper to them in a DSP framework.

C++ being the language being emphasized in this effort, the following libraries and tools will be adopted and may be considered a prerequisite.

- **GSL** - GNU Scientific Library `http://www.gnu.org/software/gsl/`
- **GNUPlot** graphing tools `http://www.gnuplot.info/`
- **GNU Octave** `https://www.gnu.org/software/octave/index`
- **libsndfile** - a library for reading and writing audio files `http://www.mega-nerd.com/libsndfile/`.

1.1 The approach

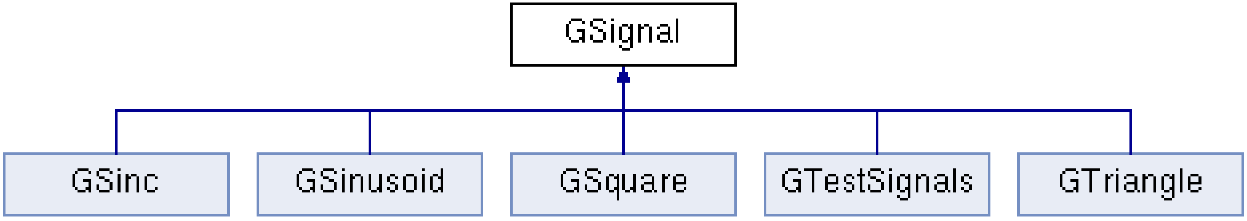
Deriving our inspiration from Prof Allen Downey and his repository `https://github.com/AllenDowney/ThinkDSP` the components herein will mirror the **Python** solutions. **GSL** will provide the toolset and **GNUPlot** or **Octave** will be exploited for plotting and illustrations. Example applications are mostly illustrative of individual features and will largely mirror the ones in the aforementioned book - using audio file support provided by **libsndfile**.

Signal generation, Spectral Analysis, Digital filters will be some areas explored in this projectlet.

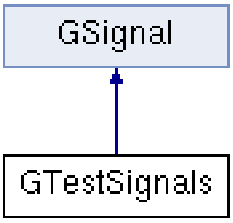
2 The library

2.1 GSignal

GSignal representing the concept of periodic signals. For example GSinusoid is a periodic signal with an amplitude, frequency and a phase offset.



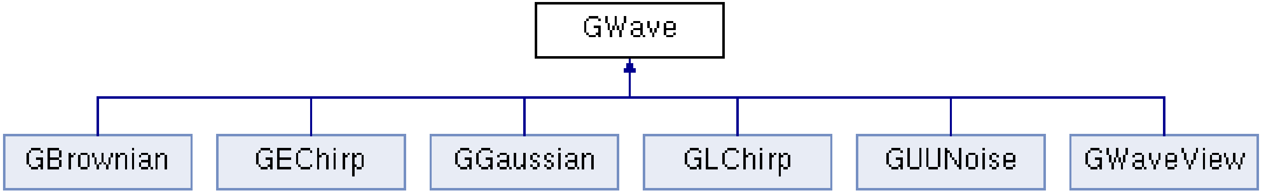
2.2 Test Signals



GTestSignals are signals that are useful for specific applications. A GStep signal for example can be used to extract a segment from another signal.

2.3 GWave

GWave representing discrete samples on the time axis. A periodic signal can be requested to generate a GWave for a time span.

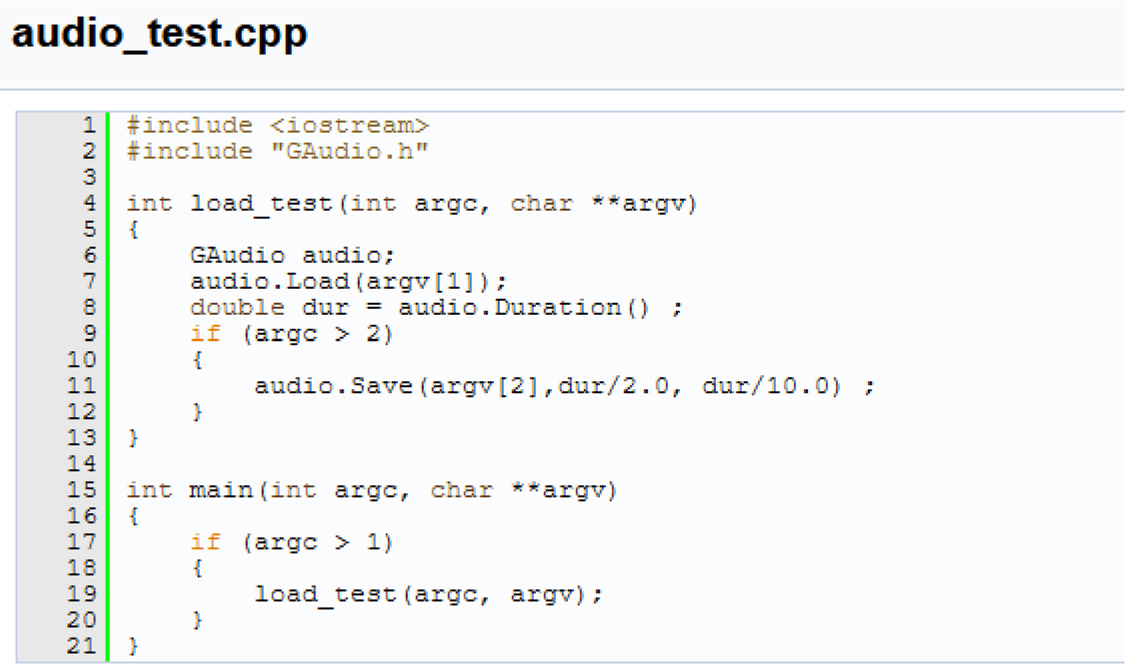


GUUNoise, GBrownian, GGaussian are some examples of non periodic waveforms.

2.4 GSpectrum

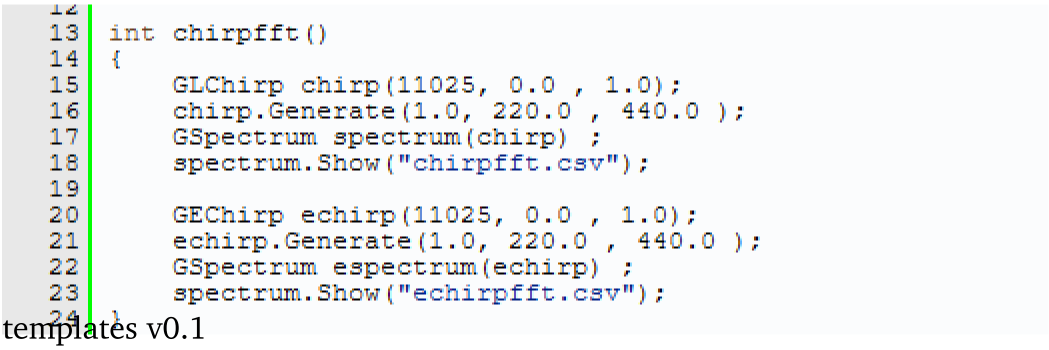
GSpectrum this is a frequency domain representation of a time series ie a GWave. A GWave can be synthesized ie converted to the time domain.

3 Example Applications



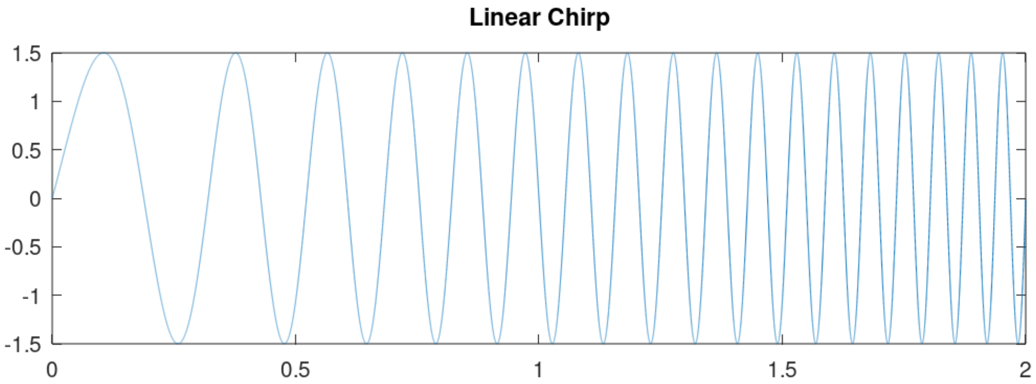
audio_test.cpp

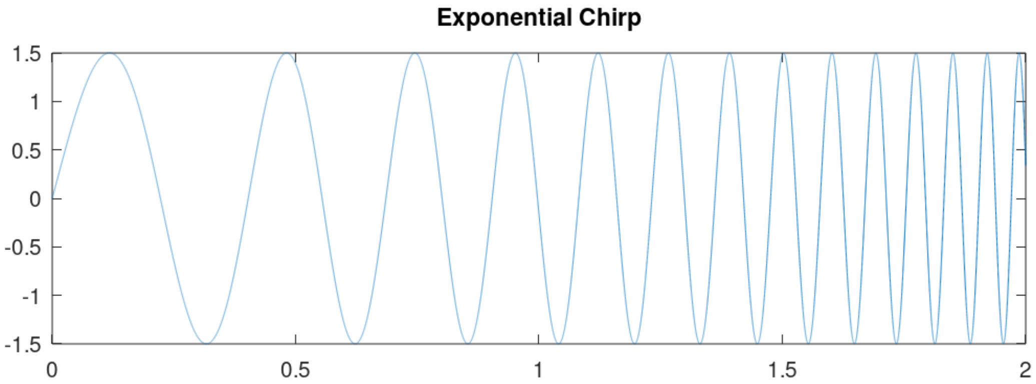
```cpp
#include <iostream>
#include "GAudio.h"

int load_test(int argc, char **argv)
{
    GAudio audio;
    audio.Load(argv[1]);
    double dur = audio.Duration() ;
    if (argc > 2)
    {
        audio.Save(argv[2],dur/2.0, dur/10.0) ;
    }
}

int main(int argc, char **argv)
{
    if (argc > 1)
    {
        load_test(argc, argv);
    }
}
```



```cpp
int chirpfft()
{
    GLChirp chirp(11025, 0.0 , 1.0);
    chirp.Generate(1.0, 220.0 , 440.0 );
    GSpectrum spectrum(chirp) ;
    spectrum.Show("chirpfft.csv");

    GEChirp echirp(11025, 0.0 , 1.0);
    echirp.Generate(1.0, 220.0 , 440.0 );
    GSpectrum espectrum(echirp) ;
    spectrum.Show("echirpfft.csv");
}
```

4 Example plots from generated signals





5 Implementation

Implementation in **C++** : `https://gitlab.com/cpp8/thinkdsp.git`